```
ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE MNEM1
OBJECT MODULE PLACED IN MNEM1.OBJ
COMPILER INVOKED BY:  :F0: MNEM1.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982)


                $title ('INSTRUCTION MNEMONICS MODULE - PART 1')
  1             mnem1:
                do;

                /*

                  modified  4/10/81  R. Silberstein
                  modified  6/16/81  R. Silberstein

                */

                /******************* INSTRUCTION MNEMONICS ******************/
                /*

                       This is all the instruction mnemonics for
                       the assembler. The mnemonics are grouped
                       according to the 6-bit hash value of the
                       mnemonics - values range from 0 to 0FH.

                       For each instruction, there is a pointer to
                       its codemacro definition.
                */

                $include (:f1:mnem.lit)
            =   $nolist
                $include (:f1:cmlink.ext)
            =   $nolist


                   /********* MNEMONICS TABLE ********/

  4   1         declare

                /*
                                                        * HASH VALUE (HEX)  *
                                                        --------------------
                */
                test    opcod4        data (0,4,.test10,'TEST'),     /* 0 */
                push    opcod4 public data (.test,4,.push3,'PUSH'),
                SCASW   OPCOD5        DATA (0,5,.SCASW1,'SCASW'),    /* 1 */
                repz    opcod4 public data (.SCASW,4,.repe1,'REPZ'),
                                                                     /* 2 */
                aaa     opcod3 public data (0,3,.aaa1,'AAA'),        /* 3 */
                                                                     /* 4 */
                movs    opcod4 public data (0,4,.movs2,'MOVS'),      /* 5 */
                daa     opcod3        data (0,3,.daa1,'DAA'),        /* 6 */
                aad     opcod3        data (.daa,3,.aad1,'AAD'),
                pushf   opcod5 public data (.aad,5,.pushf1,'PUSHF'),
                MOVSB   OPCOD5 PUBLIC DATA (0,5,.MOVSB1,'MOVSB'),    /* 7 */
                adc     opcod3 public data (0,3,.adc11,'ADC'),       /* 8 */
                stos    opcod4        data (0,4,.stos2,'STOS'),      /* 9 */
```

CP/M-86 v.1.1 (vol. IV)
COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P.O. BOX 579
PACIFIC GROVE, CA 93950
SER. # C81-162

```
            LODSW   OPCOD5        DATA (.STOS,5,.LODSW1,'LODSW'),
            add     opcod3 public data (.LODSW,3,.add11,'ADD'),
            CMPSW   OPCOD5 PUBLIC DATA (0,5,.CMPSW1,'CMPSW'),    /* 0a */
            STOSB   OPCOD5        DATA (0,5,.STOSB1,'STOSB'),    /* 0b */
            ja      opcod2 public data (.STOSB,2,.ja1,'JA'),
            jb      opcod2        data (0,2,.jb1,'JB'),          /* 0c */
            dec     opcod3 public data (.jb,3,.dec3,'DEC'),
            JC      OPCOD2        DATA (0,2,.JB1,'JC'),          /* 0D */
            loopne  opcod6 public data ( JC,6,.loopne1,'LOOPNE'),
                                                                 /* 0e */
            aam     opcod3        data (0,3,.aam1,'AAM'),        /* 0f */
            je      opcod2        data (.aam,2,.je1,'JE'),
            repnz   opcod5 public data (.je,5,.repne1,'REPNZ');

  5  1     end mnem1;
```

MODULE INFORMATION:

```
    CODE AREA SIZE     = 0000H      0D
    CONSTANT AREA SIZE = 00D2H    210D
    VARIABLE AREA SIZE = 0000H      0D
    MAXIMUM STACK SIZE = 0000H      0D
    195 LINES READ
    0 PROGRAM ERROR(S)
```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE MNEM2
OBJECT MODULE PLACED IN MNEM2.OBJ
COMPILER INVOKED BY:  :F0:MNEM2.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982)


            $title ('INSTRUCTION MNEMONICS MODULE - PART 2')
  1         mnem2:
            do;

            /*

              modified  4/10/81  R. Silberstein
              modified  6/16/81  R. Silberstein
              modified  7/24/81  R. Silberstein

            */

            /******************* INSTRUCTION MNEMONICS ******************/
            /*

                  This is all the instruction mnemonics for
                  the assembler. The mnemonics are grouped
                  according to the 6-bit hash value of the
                  mnemonics - values range from 10H to 1FH.

                  For each instruction, there is a pointer to
                  its codemacro definition.
            */

            $include (:f1:mnem.lit)
        =   $nolist
            $include (:f1:cmlink.ext)
        =   $nolist

               /********** MNEMONICS TABLE ********/

  4  1      declare

            /*
                                                    * HASH VALUE (HEX)  *
                                                    --------------------
            */
            jae     opcod3 public data (0,3,.jae1,'JAE'),       /* 10 */
            jbe     opcod3         data (0,3,.jbe1,'JBE'),      /* 11 */
            jg      opcod2 public data (.jbe,2,.jg1,'JG'),
            lea     opcod3         data (0,3,.lea1,'LEA'),      /* 12 */
            clc     opcod3 public data (.lea,3,.clc1,'CLC'),
            cmc     opcod3         data (0,3,.cmc1,'CMC'),      /* 13 */
            cld     opcod3         data (.cmc,3,.cld1,'CLD'),
            iand    opcod3 public data (.cld,3,.and10,'AND'),
            loopz   opcod5 public data (0,5,.loope1,'LOOPZ'),   /* 14 */
            aas     opcod3 public data (0,3,.aas1,'AAS'),       /* 15 */
            jge     opcod3         data (0,3,.jge1,'JGE'),      /* 16 */
            jl      opcod2 public data (.jge,2,.jl1,'JL'),
            sbb     opcod3         data (0,3,.sbb11,'SBB'),     /* 17 */
            in      opcod2 public data (.sbb,2,.in4,'IN'),
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
            das     opcod3        data (0,3,.das1,'DAS'),       /* 18 */
            cli     opcod3 public data (.das,3,.cli1,'CLI'),
            jna     opcod3        data (0,3,.jbe1,'JNA'),       /* 19 */
            jo      opcod2 public data (.jna,2,.jo1,'JO'),
            jnb     opcod3        data (0,3,.jae1,'JNB'),       /* 1a */
            jp      opcod2        data (.jnb,2,.jp1,'JP'),
            neg     opcod3        data (.jp,3,.neg2,'NEG'),
            inc     opcod3 public data (.neg,3,.inc3,'INC'),
            JNC     OPCOD3        DATA (0,3,.JAE1,'JNC'),       /* 1B */
            esc     opcod3        data (.JNC,3,.esc3,'ESC'),
            jle     opcod3        data (.esc,3,.jle1,'JLE'),
            lahf    opcod4 public data (.jle,4,.lahf1,'LAHF'),
            cbw     opcod3        data (0,3,.cbw1,'CBW'),       /* 1c */
            MOVSW   OPCOD5        DATA (.CBW,5,.MOVSW1,'MOVSW'),
            icall   opcod4 public data (.MOVSW,4,.call3,'CALL'),
            js      opcod2        data (0,2,.js1,'JS'),         /* 1d */
            jne     opcod3 public data (.js,3,.jne1,'JNE'),
            jnae    opcod4        data (0,4,.jb1,'JNAE'),       /* 1e */
            cwd     opcod3 public data (.jnae,3,.cwd1,'CWD'),
            jpe     opcod3        data (0,3,.jp1,'JPE'),        /* 1f */
            jng     opcod3        data (.jpe,3,.jle1,'JNG'),
            jnbe    opcod4 public data (.jng,4,.ja1,'JNBE');

  5  1     end mnem2;
```

MODULE INFORMATION:

```
    CODE AREA SIZE     = 0000H      0D
    CONSTANT AREA SIZE = 0122H    290D
    VARIABLE AREA SIZE = 0000H      0D
    MAXIMUM STACK SIZE = 0000H      0D
    204 LINES READ
    0 PROGRAM ERROR(S)
```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE MNEM3
OBJECT MODULE PLACED IN MNEM3.OBJ
COMPILER INVOKED BY:  :F0:MNEM3.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982)


            $title ('INSTRUCTION MNEMONICS MODULE - PART 3')
  1         mnem3:
            do;

            /*

              modified  6/16/81  R. Silberstein

            */

            /******************* INSTRUCTION MNEMONICS ******************/
            /*

                  This is all the instruction mnemonics for
                  the assembler. The mnemonics are grouped
                  according to the 6-bit hash value of the
                  mnemonics - values range from 20H to 2FH.

                  For each instruction, there is a pointer t
                  its codemacro definition.
            */

            $include (:f1:mnem.lit)
        =   $nolist
            $include (:f1:cmlink.ext)
        =   $nolist

               /********** MNEMONICS TABLE ********/

  4   1     declare

            /*
                                                  * HASH VALUE (HEX)  *
                                                  --------------------
            */
            isal    opcod3        data (0,3,.sal4,'SAL'),       /* 20 */
            STOSW   OPCOD5        DATA (.ISAL,5,.STOSW1,'STOSW'),
            cmp     opcod3 public data (.STOSW,3,.cmp11,'CMP'),
            rcl     opcod3        data (0,3,.rcl4,'RCL'),       /* 21 */
            ior     opcod2 public data (.rcl,2,.or10,'OR'),
            loopnz  opcod6        data (0,6,.loopne1,'LOOPNZ'), /* 22 */
            sahf    opcod4        data (.loopnz,4,.sahf1,'SAHF'),
            callf   opcod5 public data (.sahf,5,.callf2,'CALLF'),
            lds     opcod3        data (0,3,.lds1,'LDS'),       /* 23 */
            div     opcod3 public data (.lds,3,.div2,'DIV'),
            jnge    opcod4        data (0,4,.jl1,'JNGE'),       /* 24 */
            jnl     opcod3        data (.jnge,3,.jge1,'JNL'),
            jz      opcod2        data (.jnl,2,.je1,'JZ'),
            les     opcod3 public data (.jz,3,.les1,'LES'),
                                                                /* 25 */
            sar     opcod3 public data (0,3,.sar4,'SAR'),       /* 26 */
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
            jno     opcod3        data (0,3,.jno1,'JNO'),       /* 27 */
            rcr     opcod3        data (.jno,3,.rcr4,'RCR'),
            rep     opcod3        data (.rcr,3,.rep1,'REP'),
            ishl    opcod3        data (.rep,3,.sal4,'SHL'),
            jmp     opcod3 public data (.ishl,3,.jmp2,'JMP'),
            jnp     opcod3        data (0,3,.jnp1,'JNP'),       /* 28 */
            hlt     opcod3 public data (.jnp,3,.hlt1,'HLT'),
            jnle    opcod4        data (0,4,.jg1,'JNLE'),       /* 29 */
            jpo     opcod3        data (.jnle,3,.jnp1,'JPO'),
            lock    opcod4 public data (.jpo,4,.lock1,'LOCK'),
            scas    opcod4        data (0,4,.scas2,'SCAS'),     /* 2a */
            stc     opcod3        data (.scas,3,.stc1,'STC'),
            sub     opcod3        data (.stc,3,.sub11,'SUB'),
            xchg    opcod4 public data (.sub,4,.xchg6,'XCHG'),
            jns     opcod3        data (0,3,.jns1,'JNS'),       /* 2b */
            std     opcod3        data (.jns,3,.std1,'STD'),
            int     opcod3        data (.std,3,.int2,'INT'),
            ret     opcod3 public data (.int,3,.ret3,'RET'),
            repe    opcod4        data (0,4,.repe1,'REPE'),     /* 2c */
            SCASB   OPCOD5        DATA (.REPE,5,.SCASB1,'SCASB'),
            idiv    opcod4 public data (.SCASB,4,.idiv2,'IDIV'),
            nop     opcod3        data (0,3,.nop1,'NOP'),       /* 2d */
            rol     opcod3        data (.nop,3,.rol4,'ROL'),
            ishr    opcod3        data (.rol,3,.shr4,'SHR'),
            jmpf    opcod4 public data (.ishr,4,.jmpf2,'JMPF'),
            mul     opcod3 public data (0,3,.mul2,'MUL'),       /* 2e */
            pop     opcod3 public data (0,3,.pop4,'POP');       /* 2f */

  5  1     end mnem3;
```

```
MODULE INFORMATION:

    CODE AREA SIZE     = 0000H      0D
    CONSTANT AREA SIZE = 0160H    352D
    VARIABLE AREA SIZE = 0000H      0D
    MAXIMUM STACK SIZE = 0000H      0D
    209 LINES READ
    0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. #_____

```
ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE MNEM4
OBJECT MODULE PLACED IN MNEM4.OBJ
COMPILER INVOKED BY:  :F0:MNEM4.PLM DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982)


                $title ('INSTRUCTION MNEMONICS MODULE - PART 4')
  1             mnem4:
                do;

                /*

                  modified  6/16/81  R. Silberstein

                */

                /**************** INSTRUCTION MNEMONICS ****************/
                /*

                        This is all the instruction mnemonics for
                        the assembler. The mnemonics are grouped
                        according to the 6-bit hash value of the
                        mnemonics - values range from 30H to 3FH.

                        For each instruction, there is a pointer to
                        its codemacro definition.
                */

                $include (:f1:mnem.lit)
          =     $nolist
                $include (:f1:cmlink.ext)
          =     $nolist

                    /********* MNEMONICS TABLE ********/

  4  1          declare

                /*
                                                        * HASH VALUE (HEX)  *
                                                        --------------------
                */
                sti     opcod3 public data (0,3,.sti1,'STI'),       /* 30 */
                retf    opcod4         data (0,4,.retf3,'RETF'),    /* 31 */
                inot    opcod3 public data (.retf,3,.not2,'NOT'),
                lods    opcod4         data (0,4,.lods2,'LODS'),    /* 32 */
                jnz     opcod3         data (.lods,3,.jne1,'JNZ'),
                mov     opcod3 public data (.jnz,3,.mov17,'MOV'),
                ror     opcod3         data (0,3,.ror4,'ROR'),      /* 33 */
                cmps    opcod4 public data (.ror,4,.cmps2,'CMPS'),
                LODSB   OPCOD5         DATA (0,5,.LODSB1,'LODSB'),  /* 34 */
                iret    opcod4 public data (.lodsb,4,.iret1,'IRET'),
                wait    opcod4         data (0,4,.wait1,'WAIT'),    /* 35 */
                CMPSB   OPCOD5         DATA (.WAIT,5,.CMPSB1,'CMPSB'),
                popf    opcod4 public data (.CMPSB,4,.popf1,'POPF'),
                                                                    /* 36 */
                imul    opcod4 public data (0,4,.imul2,'IMUL'),     /* 37 */
                out     opcod3 public data (0,3,.out4,'OUT'),       /* 38 */
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____

```
            retn    opcod4        data (0,4,.ret3,'RETN'),      /* 39 */
            ixor    opcod3        data (.retn,3,.xor10,'XOR'),
            xlat    opcod4 public data (.ixor,4,.xlat1,'XLAT'),
            repne   opcod5        data (0,5,.repne1,'REPNE'),   /* 3a */
            into    opcod4        data (.repne,4,.into1,'INTO'),
            loop    opcod4        data (.into,4,.loop1,'LOOP'),
            jmps    opcod4 public data (.loop,4,.jmps1,'JMPS'),
                                                              /* 3b */
                                                              /* 3c */
                                                              /* 3d */
                                                              /* 3e */
            jcxz    opcod4        data (0,4,.jcxz1,'JCXZ'),     /* 3f */
            loope   opcod5 public data (.jcxz,5,.loope1,'LOOPE');

            /*** ********** END OF MNEMONIC TABLE ****************/

  5  1      end mnem4;
```

MODULE INFORMATION:

```
    CODE AREA SIZE     = 0000H      0D
    CONSTANT AREA SIZE = 00D5H    213D
    VARIABLE AREA SIZE = 0000H      0D
    MAXIMUM STACK SIZE = 0000H      0D
    197 LINES READ
    0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

COPYRIGHT © 1981, 1982
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #_____